



A. IDENTIFICACIÓN

CARRERA:	INGENIERÍA DE SISTEMAS
ASIGNATURA:	METODOLOGÍA DE LA PROGRAMACIÓN II
SIGLA:	SIS 2210
DURACIÓN:	Un semestre académico (20 semanas)
HORAS SEMANALES:	Teóricas: 4, Prácticas: 1, Laboratorio: 2, TOTAL: 7
PLAN DE ESTUDIOS:	2011

B. CONTRIBUCIÓN AL PERFIL

Objetivos:

A la conclusión de la asignatura, el estudiante debe ser capaz de: crear soluciones ordenadas y fácilmente comprensibles para cualquier otro programador, comprender y aplicar los conceptos de programación orientada a objetos, Desarrollar la capacidad de aplicar la programación modular y estructurada, aplicar la filosofía de la programación orientada a objetos, y realizar aplicaciones en entorno Windows

Unidades de competencia:

- Habilidad para aplicar conocimientos matemáticos científicos y de ingeniería.
- Habilidad para diseñar, conducir y controlar experimentos así como analizar e interpretar datos.
- Habilidad para identificar y solucionar problemas de ingeniería.
- Habilidad para usar técnicas, destrezas y herramientas necesarias para la práctica de la ingeniería.
- Desarrollar aplicaciones basadas en herramientas computacionales para solucionar problemas de ingeniería.

C. CONTENIDO PROGRAMÁTICO

Contenido mínimo:

Paradigma de POO.- Conceptos básicos de programación Orientada a Objetos.- Clases y Objetos.- Clases abstractas y herencia.- Polimorfismo.- Sobrecarga.- Plantillas.- Archivos.-

Contenido analítico:

Tema 1: Paradigma de programación orientada a objetos (POO)

- 1.1 Evolución
- 1.2 Beneficios
- 1.3 Desventajas

Tema 2: Conceptos básicos de programación orientada a objetos



- 2.1 Objetos
- 2.2 Clases
- 2.3 Constructores y destructores
- 2.4 Métodos
- 2.5 Atributos
- 2.6 Sobrecarga
- 2.7 Tipos de datos parametrizados
- 2.8 Herencia
- 2.9 Vinculación dinámica
- 2.10 Polimorfismo

Tema 3: Clases y objetos

- 3.1 Concepto de clase y objeto
- 3.2 Creación de clases.
- 3.3 Instancia de una clase.
- 3.4 Diagramas de clases y objetos.
- 3.5 Constructores y destructores.
- 3.6 Objetos dinámicos.
- 3.7 Estructura dinámica de datos.

Tema 4: Clases abstractas y herencia

- 4.1 Abstracción.
- 4.2 Herencia. Clases derivadas.
- 4.3 Herencia repetida y clases base virtuales
- 4.4 Funciones virtuales puras
- 4.5 Diseño de clases abstracta.

Tema 5: Polimorfismo

- 5.1 Funciones virtuales
- 5.2 Polimorfismo sin ligaduras virtuales
- 5.3 Polimorfismo con ligaduras virtuales
- 5.4 Ventajas del polimorfismo

Tema 6: Sobrecarga.

- 6.1 Sobrecarga de operadores
- 6.2 Sobrecarga de funciones
- 6.3 Aplicación de sobrecarga a diferentes tipos de datos
- 6.4 Estructuras dinámicas de datos
- 6.5 Aplicaciones de sobrecarga

Tema 7: Plantillas.

- 7.1 Conceptos generales sobre plantillas
- 7.2 Plantillas de funciones
- 7.3 Definición de plantillas de función



- 7.4 Plantillas de clases
- 7.5 Definición de las funciones miembro
- 7.6 Instanciación de una plantilla con clases
- 7.7 Aplicaciones de plantillas.

Tema 8: Archivos

- 8.1 Tipos de archivos
- 8.2 Flujos

D. REFERENCIAS BIBLIOGRÁFICAS

- [1] Bratley J. W., Tucker All, **Fundamentos de Informática** (Segunda Edición).
- [2] Deitel & Deitel. **Como Programar en C++** (Segunda Edición).
- [3] Jammes. **Algorítmica, Especificación de Algoritmos Procedurales y Funcionales** (Primera Edición).
- [4] Joyanes Aguilar, L. (1993). **Turbo Pascal 7.0** (Primera Edición).
- [5] Alcalde E., García M., (1990). **Metodología de la Programación**. Ed. McGraw-Hill.
- [6] Joyanes Aguilar L., (1988). **Fundamentos de la Programación, algoritmos y estructura de datos**. Ed. McGraw-Hill.
- [7] Zeballos J. **Programación Orientada a Objetos** (Cuarta Edición).